MULTILEVEL SECURITY WITHOUT ENCRYPTION

Mr. Michael J. Ratway

Currently, there is no multilevel security (MLS) system composed of heterogeneous networks and computers certified for operation in a ship environment. Proposed solutions to this MLS problem rely on encryption. Encryption—the transformation of plain text into cipher, which usually has the appearance of random unintelligible data—is the second step in secure data transfers, used to deny access to unwanted receivers. The first step is authentication of sender and receiver. This article offers a design for an MLS system without encryption; that is, a system that uses authentication only. The design key is to mix false information with the valid information and use authentication to separate the two. The cost is increased bandwidth, which the design strives to minimize. The savings are total software implementation requiring no encryption hardware; and no need for disparate communications.

INTRODUCTION

The white paper, "Chaffing and Winnowing: Confidentiality without Encryption,"[1] initiated the thoughts of designing an MLS system without encryption. The Chaffing and Winnowing technique described in Rivest's paper is aimed at defeating strong U.S. data-encryption technology export laws by supplying data security without encryption—just authentication. Ignoring the legal discussion, the Chaffing and Winnowing technique is applied to the design of an MLS concept in this article.

THE MLS PROBLEM DOMAIN

The purpose of any MLS is to provide a level of trust for data access at varying security levels. Consider the problem of processing unclassified and secret digital data using personnel with no clearance and those with a secret clearance. A multilevel, secure, automated data processing system allows one system to house unclassified and secret information, and administers access based on personnel clearance. Without MLS, two alternatives are commonly employed. The first alternative is called system high, where the automated data processing system is continuously operated at the highest level of data classification. The disadvantage to this solution is everyone must be cleared at the highest level to process even unclassified information. A second alternative to MLS is using separate, automated data processing systems: one for each distinct data classification level. This solution slows the transfer of information (digital data) from the lower to higher classification levels, in addition to increasing the hardware and operating costs.

Understand that it is possible to defeat any security system given resources and time. To increase the level of trust for an MLS system, information is encrypted. For adversaries, encryption increases the time required to perform a brute force attack using the best resources to several decades. With strong computer security practices in effect, adversaries are limited to physical attacks. Coercion, enlisting spies, or electromagnetic eavesdropping are typical physical security attacks. These and operating systems' exploits (e.g., screen capture programs) are common to any MLS system and are not solved in this MLS design. This design offers the use of chaffing and winnowing as an alternative to encryption for increasing the level of trust in an MLS design.

AUTHENTICATION

Any MLS system must authenticate the users of the system. Authentication is the process of validating information to be true. The most common type of computer authentication is a user entry sequence consisting of user identification (name) and a password. The password employs something known only to the user in the authentication scheme. Other methods of user authentication employ something possessed—e.g., a key or smart card—or something embodied, such as a fingerprint.[2,3]

Like user authentication, formal methods exist to authenticate digital data. Message Authentication Code (MAC) employs a one-way function to create a signature for the digital data. Two properties of one-way functions are that: it is nearly impossible to find any two distinct files that have the same digital signature, and the one-way function has no inverse. Message Digest 5 (MD5) is the widely digital data authentication mechanism in use. MD5 uses an initialization vector that allows anyone to sign or verify a message. MACs differ in that they use a shared secret key between parties instead of an initialization vector.[4] In this article, digital data authentication implies the usage of a secret shared key between parties, or MACs.

The process of authenticating a user is known as *identification*. The process of validating content and origin of digital data is known as *authentication*.

CHAFFING DESIGN

The design for an MLS system that uses authentication and chaff is simple. Sending or storing data has two parts: authenticating and adding chaff. Chaff in this context is invalid data that mirrors authentic data. In order to understand the data, a user must remove the chaff. The originator of data divides the information into parts called packets and authenticates each packet using a secret authentication key. That is, the originator appends to each packet a MAC computed as a function of the packet contents and the secret authentication key. The legitimate user, knowing the secret authentication key, can determine that a packet is authentic by computing the MAC and comparing it to the received MAC. Thus the valid user or recipient winnows the valid data from the chaff with a shared, secret authentication key and MAC algorithm.

To illustrate how authentication can be used to increase data security, consider creating a chaffing design for a switched network having a payload of 48 bytes. To ensure one chaff packet for each data packet, the 48 byte payload is split in two: 24 bytes for data and 24 bytes for chaff. Because the original data is broken into packets, serial numbers are added to the packets for reconstruction. Hence, a valid data packet has a serial number, data, and digital signature. Chaff, an invalid data packet, has the same serial number as a valid data packet, forged data, and an invalid digital signature. One possible distribution of a 48-bytes payload is provided in Table 1. The structure is repeated in Table 1 to emphasize that chaff and valid data are

Table 1—Usage of Bytes in a 48-Byte Network Packet

Number of Bytes	Usage
4	Serial Number
12	Data
8	Digital Signature
4	Serial Number
12	Data
8	Digital Signature

inserted in the buffer at random. Table 2 provides a sample network packet.

In terms of storage, the cost of securing data in this design is 400%. Forty-eight bytes of total information are used for each 12 bytes of usable data. The second problem is the generation of chaff. Anyone who knows the author's passion for ice cream and dislike for hot dogs can detect the chaff. Observe that, the creation of chaff capable of spoofing a knowledgeable adversary is nontrivial. One solution to this problem is to use an authentication function that hashes the data. An authentication function that uses hashing jumbles the input till it appears to be random data, then computes a MAC based on the random data.[4,5] By substituting the authentication function hash of the input, the original data emerges as random data. Because the valid data is randomized, chaff can be generated in a random fashion. After applying this solution, many of the characters are no longer printable: this problem is overcome by representing each character with two hexadecimal characters separated with a space; the serial number and 64-bit digital signature are represented with 8 and 16 hexadecimal characters, respectively. Using this print format, the sample network packet after hashing is shown in Table 3.

Sample source code in C of a symmetrical MAC (produces the same digital signature of the input and the jumbled input provided that the same secret key is used for both) that generated Table 3 is available. The code is for educational purposes only and has not been subject to any formal cryptanalysis.

Note that the serial number is treated as part of the data and supplied to the authentication function. Because the serial number is hashed the chaff buffer no longer carries an identical serial number. *This is transparent to a valid user and can be used to complicate the task of adversaries.* Admittedly all data hashing functions could be construed as cryptography, although hashing is not necessary in the chaffing security design. Imagine if the data is authenticated on a character-by-character basis with more than one character of chaff added for each valid character. With only a few chaff characters, it quickly becomes impossible for adversaries to reconstruct the data without the secret key to the MAC. Of course, data storage economy precludes such a design except for small data sets.

Table 2—Sample Network Packet

Serial Number	Data	MAC
0001	Like Hotdogs	544402111
0001	Eat Icecream	902019826

At this point, it is interesting to note that valid data for one user is chaff to another user. In our example, "Like Hotdogs," describes Dan and is authenticated using Dan's secret key, while "Eat Icecream" applies to Mike and is authenticated using Mike's secret key. Using Mike's secret key to authenticate, "Like Hotdogs" appears as chaff. Therefore, in an environment with multiple-like pieces of data having MACs computed with different secret keys chaff is automatically present, as in an MLS environment. The data in Table 3 was generated this way, using two distinct secret keys. The cost of achieving confidentiality, in terms of storage, is reduced to 200% for this design.

Table 3—Hashed Sample Network Packet in Hexadecimal

Serial Number	Data	MAC
4398376e	f8 66 bf 94 c1 73 1b 31 3f 5b 8d 58	FD5904C8E74A7CB8
c395ea61	99 18 73 22 42 9d 16 26 a0 49 f9 56	3B54A9874DB37FFA

The last point in the MLS using authentication design is that chaff could be added in different places: the operating system, communication card, or network device. Creating a chaff packet is simple: duplicate the packet serial number, add random data, and falsify the authentication; or authenticate data (possibly false) with a different authentication key (possibly false).

WINNOWING DESIGN

Winnowing is the processing of separating the authentic packets from false packets (chaff). Each packet (24 bytes in this case) includes a MAC for authentication with a secret key; all packets failing to authenticate are winnowed. After removal of the chaff, valid data packets are assembled in correct order using serial numbers to reconstruct the information. Then the data is transferred to the application that supplied the authentication information (secret key).

For an adversary, separating the chaff from valid packets will be proportional to the number of ways a subsequence of packets can be picked and tested as being valid; this will be exponential in the total number of packets with a sufficient number of chaff blocks.

MLS AREAS NOT ADDRESSED

TEMPEST

TEMPEST is electronic eavesdropping. All electronic equipment—hair dryers, typewriters, televisions and computers—emit electrical and electromagnetic radiation through the air or through conductors. It has long been recognized that such emanations can cause interference. Notice the Federal Communication Commission compliance statement on all modern electronic equipment. This electromagnetic leakage could be intercepted and deciphered by an adversary using relatively unsophisticated equipment. To control the leakage of electromagnetic signals, shields are used to conduct them to ground before they can escape. This specialized area of computer security is typically considered separate in a system security design and is achieved by shielding equipment, rooms, or entire buildings. For a thorough introduction to TEMPEST, see Reference 2.

Operating System

The computer operating system is the center point of digital data security. It provides the interface tools between digital data and the system user. To provide trust, the operating system must control access to resources and digital data on a user basis. Utilities such as screen capture and printing must be managed. A savvy user that monitors the print buffer can obtain unwanted access in a less trusted operating system. Although the subsequent chaffing protocol could be used in designing parts of an operating system—in particular, data at rest and data in transit—overall assurance of trust for digital data manipulation by authorized users is the responsibility of the operating system.

MLS AREAS AFFECTED BY THE DESIGN

To provide data storage protection (data at rest), an MLS system must make access decisions to data based on identification and a security profile associated with it. For example, an MLS system requires a user login sequence to identify a particular user, once the MLS identifies the user, access to files is granted based on the user's profile. Alternatively, by using the chaffing protocol, all data is authenticated and stored with chaff, and the user's access to data is restricted to data the user can authenticate.

The value in digital data is its ability to be shared. To share digital data requires the data to be moved from one location to another. For sensitive information, protection must be provided for the data in transit so the data is shared only with intended recipients and not adversaries. Encryption, chaffing, or steganography provide protection of data from eavesdroppers while in transit. All of these methods require some shared, supposedly secret, piece of information between

sender and receiver. For the chaffing protocol, the secret authentication key is the shared piece of information.

The chaffing protocol adds an additional piece of assurance to data at rest or in transit—authentication. Encryption and steganography provide secrecy, not necessarily authentication. Consider an example of steganography, where a picture has a hidden message in the low-order pixel bits. An adversary intercepts and alters the picture, and the unknowing receiver processes the hidden message. Most likely, the adversary's change produces a random message or garbage to the receiver. In a rare case, a valid (but unintended) output may be generated and acted upon by the receiver. This situation is not possible when authentication of the digital data is used.

MLS CHAFFING PROTOCOL

In conclusion, the following protocol captures the MLS Design.

Securing Data

- Break data into packets, containing a serial number and data. Serial numbers need to be sequential, though they do not need to start at a fixed number. Hash the serial number and data as one piece of information. Compute a MAC for the hashed packet using an authentication key and add it to the packet. The preference is for a MAC algorithm that hashes the data while calculating the MAC.
- Add at least one chaff packet per serial number. Using MAC and hash algorithms that look like random data, this step is accomplished by creating a packet composed of random data. Alternatively, in a multiplexing environment, chaff is added by mixing more than two data streams.
- Randomize the order of the chaff and data packets.
- Store or transmit the combined packets.

Retrieving Data

- Authenticate all packets. Packets that fail to authenticate are removed.
- Reassemble data using the serial number and removing MACs.
- Deliver data to the application supplying the authentication key.

REFERENCES

1. Rivest, Ronald L., *Chaffing and Winnowing: Confidentiality without Encryption*, MIT Lab for Computer Science, http://theory.lcs.mit.edu/~rivest/chaffing.txt, 18 Mar 1998 (rev. 27 Mar 1998).

2. Russell, Deborah and Gangemi, G.T., Sr., Computer Security Basics, O'Reilly & Associates, Inc., Sebastopol, CA 95472, Dec 1991.

3. Amoroso, Edward, *Fundamentals of Computer Security Technology*, AT&T Bell Laboratories, Prentice Hall PTR, Upper Saddle River, NJ 07458, 1994.

4. "HMAC: Keyed-Hashing for Message Authentication," *Request for Comments: 2104*, http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2104.txt, Feb 1997.

5. Schneier, Bruce, *Applied Cryptography Protocols Algorithms*, and Source Code in C, John Wiley & Sons, Inc., New York, 1994.

THE AUTHOR

MR. MICHAEL J. RATWAY

Mr. Michael J. Ratway is an employee of the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) in Dahlgren, Virginia. He started his career with NSWCDD in 1983 after receiving a master's degree in mathematics from the University of Wyoming. He earned his second master's degree in electrical engineering from Virginia Polytechnic Institute and State University in 1989. He has completed 12 semester hours of study in the Information Technology doctoral program at George Mason University. Since 1989, Mr. Ratway has been part of a team designing advanced combat demonstration systems employing network communications. In addition to information technology. Other interests include automation, tactical planning, and digital data security.